



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/612,350	07/06/2000	Ronald O'Neal Edmark	AUS000057US1	7192

7590 06/03/2003

Duke W Yee
Carstens Yee & Cahoon LLP
P O Box 802334
Dallas, TX 75380

[REDACTED] EXAMINER

KENDALL, CHUCK O

[REDACTED] ART UNIT [REDACTED] PAPER NUMBER

2122

DATE MAILED: 06/03/2003

3

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/612,350	EDMARK ET AL.	
	Examiner Chuck O Kendall	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM
 THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 02 July 2000.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) _____ is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-28 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) The proposed drawing correction filed on _____ is: a) approved b) disapproved by the Examiner.
 If approved, corrected drawings are required in reply to this Office action.
- 12) The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) The translation of the foreign language provisional application has been received.
- 15) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- | | |
|--|--|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s). _____ . |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449) Paper No(s) <u>2</u> . | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

1. This action is in response to the application filed 07/06/00.
2. Claims 1-28 have been examined.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

4. Claims 1-28 are rejected under 35 U.S.C. 102(e) as being anticipated by Berry et al.

USPN 6,539,339 B1.

Regarding claim 1 Berry anticipates, monitoring performance of a program being executed using per thread metric variables with reused kernel threads comprising:

receiving an event indication (3:35-40);
ascertaining kernel thread level profile information(3:40-47);
identifying a kernel thread, wherein the kernel thread level profile information is attributed to the identified kernel thread (3:40-47);;
determining whether the identified kernel thread has been reused (17:3-5, see recursion depth for reuse); and
updating profile information with the kernel thread level profile information based whether the identified kernel thread has been reused (fig, 22, 2216).

Regarding claim 2 the method recited in claim 1 above, wherein determining whether the identified kernel thread has been reused further comprises:

 checking a hash table for an entry of the identified kernel thread (10:55-57).

Regarding claim 3 the method recited in claim 1 above, wherein updating profile information with the kernel thread level profile information based on the identified kernel thread is reused further comprises:

 applying the kernel thread level profile information for the reused identified kernel thread to one of a previous application thread and a new application thread (fig, 22, 2216).

Regarding claim 4 the method recited in claim 2 above, wherein updating profile information with the kernel thread level profile information based on the identified kernel thread is reused further comprises:

 applying the kernel thread level profile information for the reused identified kernel thread to one of a previous application thread and a new application thread (fig, 22, 2216);

 marking the previous application thread being terminated (fig 23,2302) ; and

 associating the reused identified kernel thread with the new application thread in the hash table(fig 23,2306, and refer back to10:55-57, for hash table) .

Regarding claim 5 the method recited in claim 1 above, wherein the profile information is stored in a node for a application thread which is associated with the kernel thread level profile information (23:57-60).

Regarding claim 6 the method recited in claim 1 above, wherein ascertaining kernel thread level profile information further comprises:

 sending a request for kernel thread level profile information to an operating system kernel, wherein the operating system kernel responds to the request by (3:35-40, 10:35-45):

 receiving the request for kernel thread level profile information (10:35-45);

 accessing a processor data area containing processor level accumulated profile information (10:40-55, see collected area);

 calculating a change in processor level accumulated profile information (15:20-40,55-60, for processor level see elapsed, cum, and base times, also see processor time spent from 55-60);

accessing kernel thread level profile information held by the operating system kernel (16:25-30, see timing information for profile information); updating kernel thread level profile information held by the operating system kernel with the change in processor level accumulated profile information (15:20-40,55-60,16:25-30); and

sending the kernel thread level profile information held by the operating system kernel to a requestor (23:53-55).

Regarding claim 7 the method recited in claim 5 above further comprises:

resetting the kernel thread level profile information held by the operating system kernel (25:1-4).

Regarding claim 8 the method recited in claim 1 above, wherein updating profile information with the kernel thread level profile information based whether the identified kernel thread has been reused further comprises:

updating an information area for a current application thread based on the identified kernel thread having not been reused (17:3-5, see recursion depth for reuse).

Regarding claim 9 the method recited in claim 1 above, wherein updating profile information with the kernel thread level profile information based whether the identified kernel thread has been reused further comprises: updating an information area for a previous application thread based on the identified kernel thread having been reused (17:3-5, see recursion depth for reuse, 17:3-5).

Regarding claim 10 the method recited in claim 1 above, wherein updating profile information with the kernel thread level profile information based whether the identified kernel thread has been reused further comprises:

accessing an information area for a application thread based on the identified kernel thread(10:15-20);

updating an information area for the application thread.

Regarding claim 11 the method recited in claim 3 above, wherein the application is a Java application (7:10-15).

Regarding claim 12 see reasoning in claim 1.

Regarding claim 13 the method recited in claim 12 above, wherein the application thread is a previous application thread based on the kernel thread having been used (fig 23, 2302).

Regarding claim 14 the method recited in claim 13 above, further comprises:

identifying the application thread as being terminated based on the kernel thread having been used.

Regarding claim 15 see claim 4 for reasoning.

Regarding claim 16 the method recited in claim 12 above, wherein determining if the kernel thread has been used further comprises: comparing an identity of the kernel thread to a list of identities of previously used kernel threads (fig 23, 2306).

Regarding claim 17 the method recited in claim 12 above, wherein the value of a metric variable for a kernel thread is a change in value of the metric variable since the last receipt of the metric variable for the kernel thread (fig 23,2312).

Regarding claim 18 the method recited in claim 12 above, wherein the metric variable relates to one of allocation bytes, allocation objects, time, live object and live bytes (for allocation bytes see bytes in claim 1).

Regarding claim 19 see claim 1 for reasoning.

Regarding claim 20 the method recited in claim 19 above, wherein the plurality of values of a metric variables for a plurality of kernel threads are received from an operating system kernel wherein each of the plurality of values was contained in a linked list of entries, each entry in the linked list being a value of a metric variable for a specific kernel thread (23:55-60, see link list).

Regarding claim 21, see claim 1 for reasoning.

Regarding claim 22 the system recited in claim 21 above, wherein the application thread is a previous application thread based on the kernel thread having been used (fig 23,2302).

Regarding claim 23 see reasoning in claim 4.

Regarding claim 24 the system recited in claim 21 above, wherein the application thread is a current application thread based on the kernel thread having not been used (fig 23,2302).

Regarding claim 25 the system recited in claim 21 above, wherein the determining means for determining if the kernel thread has been used further comprises:

comparing means for comparing an identity of the kernel thread to a list: of identities of previously used kernel threads (fig 23,2306, see computing difference for comparing).

Regarding claim 26. The system recited in claim 21 above, wherein the value of a metric variable for a kernel thread is a change in value of the metric variable since the last receipt of the metric variable for the kernel thread (fig 23, 2312).

Regarding claim 27 see reasoning in claim 18.

Art Unit: 2122

Regarding claim 28 see claim 1 for reasoning.

Correspondence Information

6. Any inquiries concerning this communication or earlier communications from the examiner should be directed to Chuck O. Kendall who may be reached via telephone at (703) 308-6608. The examiner can normally be reached Monday through Friday between 8:00 A.M. and 5:00 P.M. est.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, *Greg Morse* can be reached at (703) 308-4789.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

For facsimile (fax) send to 703-7467239 official and 703-7467240 draft

Chuck O. Kendall

*Software Engineer Patent Examiner
United States Department of Commerce*


GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100